

# An Equal Excess Negotiation Algorithm for Coalition Formation

Hrshikesh J. Goradia  
Department of Computer Science and  
Engineering  
University of South Carolina  
Columbia, SC 29208, USA  
goradia@sc.edu

Jose M. Vidal  
Department of Computer Science and  
Engineering  
University of South Carolina  
Columbia, SC 29208, USA  
vidal@sc.edu

## ABSTRACT

Coalition formation is an important form of interaction in multiagent systems. It enables the agents to satisfy tasks that they would otherwise be unable to perform, or would perform with a lower efficiency. The focus of our work is on real-world application domains where we have systems inhabited by rational, self-interested agents. We also assume an environment without any trusted central manager to resolve issues concerning multiple agents. For such environments, we have to determine both an optimal (utility-maximizing) coalition configuration and a stable payoff configuration, concurrently and in a distributed fashion. Solving each of these problems is known to be computationally expensive, and having to consider them together exacerbates the problem further. In this paper, we present our Progressive, Anytime, Convergent, and Time-efficient (PACT) algorithm for coalition formation to address the above concerns. We assess the stability of the resulting coalition by using a new stability concept, the *relaxed core*, which is a slight variation on the core. We show experimentally that our algorithm performs admirably in comparison to an optimal solution, it typically produces solutions that are relaxed-core-stable, and it scales well.

## 1. INTRODUCTION

Coalition formation is an important cooperation method in multiagent systems where the autonomous agents cannot accomplish individual tasks in their entirety (or accomplish them poorly) by themselves, and where there can be no central agency for coordinating the actions of all the agents. Works in game theory and microeconomics have provided many solution concepts for stable coalition formation. While many of these concepts involve a highly expensive combinatorial search to compute, game theorists have devised transfer schemes for many concepts, which describe the bargaining process that can yield a stable solution for any characteristic form game. Coalition formation research in multiagent

systems have provided both types of solutions: algorithms that yield optimal allocations but are computationally intensive [6, 2], and algorithms that produce results in polynomial time without any guarantees of optimality [7, 5]. While some work focuses on cooperative agents, thereby concerning only on increasing the group's global utility, others address self-interested agents where the stability of the ensuing solutions are of primary importance too. Our research is geared towards devising practical solutions for real-life environments with self-interested agents and no central authority. Note that although combinatorial auction research [8, 1] also targets forming buyer/seller coalitions treating them as self-interested entities, the offered solutions are all inherently centralized and require a trusted entity that aggregates the agent preferences and determines the final output. Such solutions are inappropriate for our target domains, which lack any central entity, let alone a trusted one.

## 2. PROBLEM DESCRIPTION

Given a set of  $m$  tasks  $T = \{t_1, \dots, t_m\}$  and a set of  $n$  selfish, rational agents  $A = \{a_1, \dots, a_n\}$  with their capabilities, we consider situations where each task must be addressed by a group of agents that will perform the task. Each task  $t_i \in T$  is defined as a pair  $\langle A_i, v(t_i) \rangle$ , associated with a set of agents  $A_i \subseteq A$  that need to collaborate by forming a coalition in order to complete the task successfully, and a value  $v(t_i) \in \mathfrak{R}$  which is the payoff that must be distributed amongst the agents in  $A_i$  at the completion of  $t_i$ . Our problem can be modeled as a characteristic function game with  $n$  agents, and a characteristic function  $v(S) \rightarrow \mathfrak{R}$  for each  $S \subseteq A$ . The worth of the tasks represent the coalition values of the associated agent set.

The outcome of the problem is represented as a coalition configuration,  $\bar{C}$ , which is a vector of tasks that were successfully assigned to the associated agent coalitions. Each element  $C_i \in \bar{C}$  is a tuple  $\langle t_i, A_i, \bar{u}_i \rangle$ , where  $t_i$  represents the task,  $A_i$  is the set of member agents, and  $\bar{u}_i$  is the payoff distribution vector, which defines the utility earned by each member by successfully completing the task. Here,  $\sum u_i = v(t_i)$ . We assume that an agent can participate in only one coalition at a time, so the coalition structure  $CS$  is a partition of agents  $A$  into  $|\bar{C}|$  disjoint sets. We strive to maximize the social welfare of all the agents in  $A$  by finding a coalition structure  $CS^* = \operatorname{argmax}_{CS \in M} V(CS)$ , where  $M$  is the set of all valid coalition structures, and  $V(CS) = \sum_{S \in CS} v_S$ .

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'07 May 14–18 2007, Honolulu, Hawai'i, USA.  
Copyright 2007 IFAAMAS.

### FIND-COALITION-PACT( $i$ )

```

1   $t \leftarrow 0$ 
2   $L_i \leftarrow \{c \in T : i \in c^{agents}\}$ 
3   $N^{agents} \leftarrow \bigcup_{c \in L_i} c^{agents}$ 
4  for  $c \in L_i$ 
5      do  $E^t(i, c) \leftarrow \frac{c^{value}}{|c^{agents}|}$ 
6           $fixed(c) \leftarrow FALSE$ 
7   $t \leftarrow t + 1$ 
8  for  $c \in L_i$ 
9      do  $A^t(i, c) \leftarrow \max_{c' \in L_{i-c}} E^t(i, c')$ 
10     for  $j \in c^{agents}$ 
11         do send  $A^t(i, c)$ ; receive  $A^t(j, c)$ 
12     if  $fixed(c) = FALSE$ 
13         then  $E^t(i, c) \leftarrow A^t(i, c) + \frac{c^{value} - \sum_{j \in c^{agents}} A^t(j, c)}{|c^{agents}|}$ 
14         if  $(E^t(i, c) \text{ has converged}) \vee (E^t(i, c) \leq 0)$ 
15             then  $fixed(c) \leftarrow TRUE$ 
16 if  $\exists c \in L_i : fixed(c) = FALSE$ 
17     then goto 7
18  $c_i^* \leftarrow \operatorname{argmax}_{c \in L_i} E^t(i, c)$ 
19 for  $j \in A$ 
20     do send  $c_i^*$ ; receive  $c_j^*$ ; put all into  $C_{preferred}^*$  set
21  $c^* \leftarrow \{c_x \in C_{preferred}^* : \forall x \neq y, c_x^{value} > c_y^{value}\}$ 
22 if  $i \in c^{*agents}$ 
23     then return  $c^*$ 
24  $C_{preferred}^* \leftarrow C_{preferred}^* - c^*$ 
25 for  $j \in c^{*agents}$ 
26     do  $L_i \leftarrow L_i - \{c \in L_i : j \in c\}$ 
27 if not empty?  $L_i$ 
28     then goto 18
29 return

```

Figure 1: The FIND-COALITION-PACT algorithm to find the best task allocation for agent  $i$ .

## 3. THE PACT ALGORITHM

The Progressive Anytime Convergent Time-efficient (PACT) algorithm is a greedy, distributed algorithm that facilitates selfish, rational agents to form coalitions in order to handle tasks. It is an iterative algorithm that progressively produces a better solution over time. It is also an any-time algorithm, so it can be halted at any time before its normal termination to provide the best solution found until that time. We will show in our experimental evaluation of the algorithm that it always converges and is also very efficient. It is based on the Equal Excess Theory [3, 4] for coalition formation, which is a prescriptive bargaining solution concept to realize stable solutions for any characteristic function game. We compare the PACT solutions with an axiomatic solution concept - the relaxed-core, which we describe in the next subsection. Figure 1 describes our PACT algorithm.

The algorithm starts with a pre-negotiation phase (lines 1-6) where all agents are initialized. For each coalition that an agent participates in, it initially expects an equal division of the total value of the coalition. During the negotiation phase (lines 7-17), an agent concurrently participates in all of its coalitions. For each coalition, the agent determines all the valid alternative coalitions that it can participate in. It sets the maximum expectation that it gets from any of these alternative coalitions as its logical claim from this

coalition in the following round of negotiations. Upon determining its claim for all its coalitions in the current round of negotiations, the agent exchanges its demands with all its neighbors and then computes its actual payoffs for the various coalitions. The excess is split equally between all the members and added to the agent's claim to get its actual payoff for the concerned coalition in the current negotiation round. If the agent payoff for a coalition converges or goes negative (i.e. the agent has to pay to the coalition in order to participate in it), then the agent fixes the coalition and no more negotiations are conducted for this coalition. We have noticed in our simulations that sometimes the agent payoff does not converge. Instead, it keeps toggling between two values. However, even under such unstable conditions the total excess generated within the coalition converges. Our agents test this convergence to determine the termination of the negotiation process. The post-negotiation phase (lines 18-29) involves the selection of the most preferred tasks to be performed by the agents. Each agent initially ranks all their coalitions based on their derived payoff from each of them and broadcasts its most preferred coalition. Thus, every agent is informed about each other's preferences over the tasks to be allocated. Each agent sorts this list of preferred coalitions based on their coalition values. Ties are broken by ranking the coalitions that have fewer agents ahead of the larger ones, as this can potentially maximize  $V(CS)$ . For further ties, the task that arrived in the environment earlier is preferred. Thus, every agent knows the winning coalition in the current round. The agent joins the winning coalition if it is a member, or removes all coalitions involving agents in the winning coalition and repeats the winner determination process if it is not. An agent remains unallocated if it runs out of candidate coalitions to participate in.

## 3.1 The Relaxed Core Solution Concept

DEFINITION 1. An outcome  $\bar{C} = \langle t, A', \bar{u} \rangle$  is in the **relaxed core** if there is no task  $t' = \langle A^t, v(t') \rangle$  such that  $v(t') > \sum_{i \in A^t} \bar{u}(i)$

This axiomatic solution concept ties stability directly to the tasks available. Agents can only negotiate with the agents that they cooperate with to solve an existing task<sup>1</sup>.

## 4. EXPERIMENTAL EVALUATION

We have conducted a series of simulations to test the quality of the PACT solutions along various dimensions.

In our first set of experiments, we test the performance of PACT against that of a utility-maximizing algorithm with respect to the coalition structures that the two algorithms propose. We test for various distributions of agents and tasks - the agent/task ratios range from 0.25 to 4. Our tests comparing the coalition structures in terms of the number of completed tasks show that the PACT algorithm produces solutions that are within 4% of the utilitarian solutions. When the proposed coalition structures were compared based on

<sup>1</sup>Note that the alternate approach of simply setting  $v(S) = 0$  for all subsets for which there is no task and using the standard core concept fails because it might induce different coalition structures. That is, a value of 0 is not the same as the impossibility of forming such a coalition (similarly, neither is a value of  $-\infty$  as that would rule out all coalition structures using that coalition)

the accrued total revenue, we found that the PACT solutions were within 3% of the utilitarian solutions. This suggests that the tasks that PACT allocates are mostly identical to those selected by the utilitarian solution.

Our second set of experiments test the scalability of the PACT algorithm. We first test PACT scalability in terms of the number of agents and tasks for various agent/task distributions. Our results show that the negotiations range roughly from 10 to 275 rounds based on agent connectivity. When there are very few agents in the system, each agent will be highly connected to all the tasks in the environment, and therefore will have similar negotiating power in the bargaining process. Due to the lack of competition, these agents quickly converge to the task(s) that offer them maximum benefits. On the other hand, when we have too many agents in the system, then their connectivity to the tasks in the environment, and therefore their negotiating power, will be very limited. Again, such agents greedily accept their best offers and converge quickly. When there are just enough agents in the system such that they are all heavily connected, and they also compete with each other for the best tasks as not all of them can be supported in the most coveted ones, the negotiations prolong for a long period of time. We also tested PACT scalability in terms of the coalition sizes to determine its performance in an environment where all the tasks have the same number of agents forming coalitions to perform them. Our results show that it gets progressively harder to choose tasks for allocation with increasing coalition size, since more agents have to endorse a particular coalition for it to be selected. For each agent/task distribution, we see that the smallest coalitions converge fast, while the larger ones either don't converge, or do so after extensive negotiations only.

Given that the negotiations can extend for hundreds of rounds, our next set of experiments focus on ascertaining the suitability of PACT for real-world applications. The most significant observation in these results is that PACT converges up to 90% of its best solution within the first 10 rounds of negotiations. This is true even for the most competitive environments. The quality of the solution keeps improving as we proceed with the negotiations, with a solution 95%, 98%, and 99% close to its best arrived by approximately the 20<sup>th</sup>, 30<sup>th</sup>, and 50<sup>th</sup> rounds respectively. It is only the final 1% that holds back the vast majority of the PACT runs from finishing earlier. This feature in PACT whereby it converges to a close approximation of its best solution very early in the negotiations is critical for its application in various real-world domains.

Our final set of experiments test the PACT solutions in terms of our relaxed-core stability concept. We test various agent/task distributions and coalition sizes to determine the percentage of PACT solutions that lie in the relaxed-core. Note that just like the core, the relaxed-core can also be empty for a particular game. The likelihood for the relaxed-core being empty increases with higher values for the number of tasks, agents, and coalition sizes as the heavy constraints that the solution concept demands from a game get harder to satisfy. Our results show that for 90% of settings with just a handful of agents and tasks, the PACT algorithm leads to an equilibrium solution that is relaxed-core-stable. Not surprisingly, these percentages go down with increasing number of tasks, agents, and coalition sizes.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced PACT - our fully distributed, progressive, anytime algorithm for coalition formation, that addresses both coalition structure generation and payoff distribution among selfish agents simultaneously. Our simulations show that for most cases, PACT generates coalition structures and maximizes social welfare to the extent that the optimal solution does. Being a greedy algorithm, PACT is very efficient, and we show experimentally that it scales very well with respect to the number of agents, the number of tasks, and the coalition size. The PACT algorithm extends to larger number of negotiation rounds in competitive settings, but the negotiations are extended only at the latter stages of the algorithm where it is very close to convergence. Therefore, PACT is very applicable to many real-life domains. We also introduce a new stability concept named *relaxed-core*, which is a slight variation of the core. It constrains the requirement of checking every possible agent subset for determining conflict allocations in core to checking for only the agent subsets formed by unallocated coalitions in our environment. We show experimentally that typically PACT solutions are relaxed-core-stable.

In this work, agents and tasks in the environment are known *a priori*. We would like to extend this work for open, dynamic environments like the Internet. In our current approach, we assume that the agents will truthfully reveal their highest expectations from alternative coalitions while claiming their share of the coalition value in a PACT negotiation round. In the future, would like to focus on the mechanism design issues of providing incentives to these agents such that truth-telling becomes their dominant strategy.

## 6. REFERENCES

- [1] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.
- [2] V. D. Dang and N. Jennings. Generating coalition structures with finite bound from the optimal guarantees. In *Proceedings of the Third International Joint Conference on Autonomous Agents and MultiAgent Systems*, pages 564–571. ACM, 2004.
- [3] J. P. Kahan and A. Rapoport. *Theories of coalition formation*. L. Erlbaum Associates, 1984.
- [4] S. S. Komorita. An equal excess model for coalition formation. *Behavioral Science*, 24(6):369–381, November 1979.
- [5] S. Kraus, O. Shehory, and G. Taase. The advantages of compromising in coalition formation with incomplete information. In *Proceedings of the Third International Joint Conference on Autonomous Agents and MultiAgent Systems*, pages 588–595. ACM, 2004.
- [6] T. Sandholm, K. Larson, M. Anderson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1-2):209–238, 1999.
- [7] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, May 1998.
- [8] J. Yamamoto and K. Sycara. A stable and efficient buyer coalition formation scheme for e-marketplaces. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 576–583, Montreal, Canada, 2001. ACM Press.